REMARKS

This communication is responsive to the Office Action dated October 6, 2003. In that Office Action, claims 1 - 33 were examined and rejected under 35 U.S.C. §103(a). Reconsideration of the application is respectfully requested in light of the following remarks.

Rejection of the Independent Claims under 35 U.S.C. §103(a)

The Examiner rejected all the independent claims (i.e., 1, 11, 14, 24, 29 and 31) under 35 U.S.C. §103(a) as being unpatentable over Bush et al. (U.S. Pat. No. 6,308,319, hereinafter "Bush") in view of A. T. Garthwaite (U.S. Pat. No. 6,185,581, hereinafter "Garthwaite").

In order to establish *prima facie* obviousness under 35 U.S.C. §103(a), three basic criteria must be met, namely: (1) there must be some suggestion or motivation to combine the references or modify the reference teaching; (2) there must be a reasonable expectation of success; and (3) the reference or references when combined must teach or suggest each claim limitation (Manual of Patent Examining Procedure 2142). Applicants submit that the Office Action failed to state a *prima facie* case of obviousness, and therefore the burden has not properly shifted to Applicants to present evidence of nonobviousness.

Applicants respectfully assert that the Examiner has failed to establish a *prima facie* case of obviousness because the references fail to disclose or suggest all of the limitations of the pending claims. Specifically, Applicants assert that none of the references disclose any of the compact garbage collection tables as claimed by the Applicants. Additionally, the specific limitations related to each table are also not shown in the references.

Compact Garbage Collection Tables

The MPEP 2111.02 requires that "[a]ny terminology in the preamble that limits the structure of the claimed invention must be treated as a claim limitation." However, it appears the Examiner reviewed the prior art looking for any tables that might possibly read on the tables claimed without recognizing that the tables claimed are **compact garbage collection tables**. Thus, the Examiner apparently cites a register in Bush as an example of unique descriptor table, and a stack or mutator thread in Bush as an example of a call site table. None of the Bush references cited by the Examiner disclose any garbage collection tables. Therefore, as each of

the independent claims 1, 11, 14, 24, 29 and 31 contain similar preamble limitations clearly indicating that the call site table, the final descriptor table and the descriptor reference table are compact garbage collection tables, Bush does not anticipate this limitation of the independent claims.

Call Site Table

In the Examiner's rejection of claim 1, the Examiner cited Bush, Col. 4, lines 48-57 as evidence of "generating a first call site table storing call site identifiers" (Applicants' element 1 of claim 1) as part of "a computer process for building compact garbage collection tables adapted for use in reclaiming memory from a heap during runtime" as required by the preamble. In general, Bush is related to safely suspending the executing code so that garbage collection can be performed. Garbage collection techniques are ancillary to the Bush method. That noted, the cited section from Bush is reproduced below:

> If predefined safe points are utilized, attractive safe points must be identified and a mechanism for suspending a mutator thread (or threads) at such safe points is needed. When a thread is suspended, all the stack frames except the most recently entered (i.e., all the calling functions in the calling hierarchy) must be at call sites, hence every call site must be a safe point. Therefore, mutator code should include information describing the registers and stack locations, if any, containing pointers at call sites. Bush, Col. 4, lines 48-57

While the cited section from Bush does include a reference to call sites, it only states that every call site is a "safe point". The above section of Bush describes including information in the mutator code describing safe points which may include call sites as well as other types of safe points (See Bush, Col. 4, lines 58-60). There is no description of what this information may be, what form it may take, or how it is used by the mutator code. Furthermore, safe points are not analogous to call sites as Bush clearly states. Lastly, this discussion is unrelated to a computer process for **building compact garbage collection tables** adapted for use in reclaiming memory from a heap during runtime. Thus, Bush does not disclose generating a first call site table storing call site identifiers as part of a computer process for building compact garbage collection tables adapted for use in reclaiming memory from a heap during runtime as claimed in claims 1, 11, 14, 24, 29 and 31.

Final Descriptor Table

The Examiner cites Bush FIG. 1 and Col. 4, lines 18-25 as disclosing generating a final descriptor table storing a set of unique descriptors. The citation is provided below:

> A portion of address space 101 is dynamically allocable as heap 102. Individual memory objects (e.g., objects 150, 151, 131, 132, 133, 134 and 141) are dynamically allocated from heap 102 and, in the referencing graph of FIG. 1, are currently reachable by a set of pointers including local variable L1, external or outer variable E1, and contents of registers R3 and R6 of register context 120. Bush, Col. 4, lines 18-25.

The above section contains no reference to a table and certainly no reference to a table of unique descriptors. An inspection of FIG. 1 shows a standard computing environment of memory objects including a stack and registers with pointers into a heap. There is nothing in FIG. 1 that is specifically related to garbage collection. Note that neither the stack nor the registers in this environment are required to include unique identifiers, nor are the stack or registers "**compact garbage collection tables** adapted for use in reclaiming memory from a heap during runtime" as required in the preamble of claim 1. In fact, Bush states (see Col. 4, lines 8-10) that the environment of FIG. 1 solely relates to an executing mutator thread, not garbage collection.

It should also be noted that none of the elements described by Bush in the section are required to store a set of unique descriptors. As discussed in the Applicants' background, it is common for registers, stacks and objects in heaps to contain multiple non-unique descriptors. Nothing the Bush reference indicates that the Bush computing environment now requires stacks, registers and heaps to contain unique sets of pointers. Thus, the Bush reference does not anticipate the final descriptor table as claimed in claims 1, 11, 14, 24, 29 and 31 as it fails to disclose all the limitations of that element.

Descriptor Reference Table

The Examiner cites Garthwaite FIG. 6, Col. 7, lines 16-25 and Column 14, lines 63-67 as disclosing generating a descriptor reference table mapping a call site identifier in the first call site table to one of the unique descriptors in the final descriptor table. The citations are provided below:

FIG. 6 depicts the various generations as being divided into smaller sections, known for this purpose as "cards." Card tables 64, 66, and 68 associated with respective generations contain an entry for each of their cards. Each card-table entry includes some summary of its associated card's intergenerational-reference content. This summary may be no more than a binary indication of whether such a reference exists, but it preferably includes a list of offsets that indicate such references' locations in the card. Garthwaite, Col. 7, lines 16-25.

Further pointers 104 and 105 point to the locations in the heap at which the associated car section begins and ends, whereas pointer 106 points to the place at which the next object can be added to the car section. Garthwaite, Col. 14, lines 63-67.

The card tables cited by the Examiner include "some summary of its associated card's intergenerational-reference content" and may include an indication of "such references' locations within the card." The card table at most contains a list of locations within the card (i.e. locations within memory as the card is itself just a subsection of memory within the heap, see Garthwaite, Col. 6, lines 41-44). Therefore, the card table contains a list of locations in a subsection of a heap. Thus, the card table does not map identifiers in one compact garbage collection table to descriptors in another compact garbage collection table and Garthwaite does not anticipate this limitation of the independent claims. In fact, the card tables disclosed by Garthwaite are simple lists and perform no mapping functions whatsoever.

Conclusion

For the reasons cited above, Applicants believe that neither Bush nor Garthwaite alone or in combination anticipate all of the limitations of independent claims in the pending application. Therefore, Applicants also believe that all of the claims in the present application are in a condition for allowance and that the Examiner's rejection of the dependent claims are therefore moot.

Claims 1-33 remain pending in this application. This response is believed to be fully responsive to all points in the Office Action. Pending claims 1-33 are believed to be in a condition for allowance. In view of the above remarks, Applicant respectfully requests a Notice of Allowance. If the Examiner believes a telephone conference would advance the prosecution of this Application, the Examiner is invited to telephone the undersigned at the below-listed telephone number.

Respectfully submitted,

12/11/03
Date

Customer No.: 23552

George C. Lewis, Reg. No. 53,214
Merchant & Gould P.C.
P.O. Box 2903
Minneapolis, MN 55402-0903
(303) 357-1639
(303) 357-1671 (fax)